Rapid Beam Training at Terahertz Frequency with Contextual Multi-Armed Bandit Learning

Son Dinh-Van*, Yuen Kwan Mo†, Jasmine Zidan‡,
Daniel S. Fowler §, Alex Evans ¶, Matthew D. Higgins‖
Warwick Manufacturing Group, The University of Warwick,
Coventry, United Kingdom
Email: {*son.v.dinh, †tony.mo, ‡jasmine.zidan.1,
§dan.fowler, ¶a.evans.7, ‖m.higgins}@warwick.ac.uk

Van Chien Trinh
School of Information & Communication Technology,
Hanoi University of Science and Technology,
Hanoi, Vietnam
chientv@soict.hust.edu.vn

***Abstract*—Terahertz (THz) frequency technology holds great promise for enabling high data rates and low latency, essential for manufacturing applications within Industry 4.0. To achieve these, beam training is necessary to enable MIMO communications without the need for explicit channel state information (CSI). In this context, the Multi-Armed Bandit (MAB) algorithms are able to facilitate online learning and decision-making in beam training, eliminating the necessity for extensive offline training and data collection. In this paper, we introduce three algorithms to investigate the applications of MAB in beam training at Terahertz frequency: UCB, Loc-LinUCB, and Probing-LinUCB. While UCB builds upon the well-established Upper Confidence Bound algorithm, Loc-LinUCB and Probing-LinUCB utilize the location of the user equipment (UE) and probing information to enhance decision-making, respectively. The beam training protocols for each algorithm are also detailed. We evaluate the performance of these algorithms using data generated by the DeepMIMO framework, which simulates abrupt changes and various challenging characteristics of wireless channels encountered in realistic scenarios as UEs move. The results illustrate that Loc-LinUCB and Probing-LinUCB outperform UCB, showing the potential of leveraging contextual MAB for beam training in Terahertz communications.**

***Index Terms*—Beam training, beam tracking, bandit learning, contextual bandit, THz communications.**

I. Introduction

The Industry 4.0 has opened a new era of smart manufacturing with the integration of advanced technologies to enhance productivity and efficiency. In this context, the demand for high-speed data communication is rapidly increasing, driven by the exponential growth in mobile and wireless devices, as well as the emergence of new applications and services that require stringent data rates and low latency. Within this paradigm, Terahertz (THz) communication, which will be an important part of the sixth generation (6G) networks, shows the potential to revolutionize smart manufacturing in the Industry 4.0 [1]–[3]. The THz frequency band, spanning from 0.1 to 10 THz, offers a vast amount of available spectrum and provides the potential for addressing the connectivity needs of Industry 4.0. It enables smart manufacturing systems to support data-intensive applications, including ultra-high-definition video streaming, augmented reality-enhanced maintenance and training, and real-time quality control. Moreover, THz communication can alleviate the shortage of frequency spectrum in the sub-6 GHz and mmWave bands that have traditionally underpinned wireless communication in manufacturing environments.

Despite these potential advantages, THz communication also poses significant technical challenges that must be overcome. The high attenuation, narrow beamwidth, and susceptibility to atmospheric absorption of THz signals make it difficult to transmit them over long distances or through obstacles [1], [2]. This requires the development of new antenna technologies, signal processing techniques, and communication protocols that can overcome these challenges. One method is to use an antenna array, which can focus the signals toward the receiver in a very narrow beam to overcome the high attenuation of THz signals [2]. Nevertheless, to utilize an antenna array effectively, it is essential to design an efficient beam training algorithm that can accurately identify the optimal beamforming direction with a low complexity.

In the context of THz communication, it is essential to recognize that traditional estimation techniques may not be practical due to the high computational complexity required for large-scale array operations. Furthermore, unlike the sub-6 GHz frequencies, transmitting pilot signals over omni-directional directions results in significant path loss at THz frequencies, making it difficult for the receiver to detect them efficiently [2]. The most straightforward approach to this problem is to use exhaustive search beam training, which involves transmitting reference signals (RSs) and testing all possible transmit and receive beamforming vectors available in the codebooks of the transmitter and receiver to identify the optimal beam pair. However, this method can cause significant latency if the number of antennas is large. Recent research has explored alternative approaches to beam training using deep learning and online learning techniques. Regarding the first approach, deep neural networks (DNN) have been used extensively to learn the pattern between the wireless environment and optimal configurations [4]. For example, the research proposed in [5], [6] focused on utilizing DNNs to

This work was supported in part by the WMG Centre High Value Manufacturing Catapult, University of Warwick, Coventry, U.K.

explore the relationship between the mmWave environment and optimal beam-direction . However, this approach requires a significant amount of training data, which can be expensive, time-consuming, and raises privacy and security concerns. As an alternative, Multi-Armed Bandit (MAB) learning has been adopted as an online decision-making technique for beam training since it eliminates the need for data collection and training, making it a promising approach.

Several algorithms based on MAB were proposed for beam alignment, including the unimodal beam alignment (UBA) algorithm, which reduces the search space for optimal beam directions by leveraging the correlation between adjacent beams and the unimodal distribution of received signal power [7]. The exponential weights algorithm for exploration and exploitation (EXP3) was used in beam-alignment algorithms to cope with unpredictable environments [8]. A beam-alignment method for vehicular communications in the millimeter-wave frequency range was proposed, which leverages the directional arrival information of the vehicle as contextual data [9]. Despite providing useful insights, the approaches introduced in [7], [9] operarate based on a central node while the performance evaluation in [8] was not realistic because it does not consider the impact of surrounding objects which might lead to abrupt changes in the wireless channel.

The purpose of this study is to explore the potential use of MAB learning in beam training for THz communications in a realistic environment showing the impact of surrounding objects and abrupt changes. In particular, we propose three algorithms: UCB, Loc-LinUCB, and Probing-LinUCB. UCB is based on the well-established Upper Confidence Bound algorithm, while Loc-LinUCB and Probing-LinUCB utilize the contextual linear UCB technique which uses the side information to improve the online decision making process. To be specific, Loc-LinUCB exploits the location information of the user equipment (UE), whereas Probing-LinUCB incorporates probing data as contextual information to enhance decision-making. These algorithms can be implemented at either BS or UE, eliminating the need for a central node. We also provide detailed protocols for each algorithm and discuss their effectiveness. Additionally, the spectral efficiency achieved by these algorithms will be benchmarked using realistic data generated by the DeepMIMO framework [10]. The results indicate that incorporating side information using contextual MAB can significantly enhance the spectral efficiency performance.

Notation: Through this paper, we use lowercase and uppercase boldface letters to represent vectors and matrices, respectively. The notations $(.)^{\mathrm{T}}$ and $(.)^{\mathrm{H}}$ represent the transpose and conjugate-transpose operator, respectively. In addition, $\mathcal{CN}(0,\sigma^2)$ stands for circularly symmetric complex Gaussian distribution with zero mean and variance σ^2 while $\mathbb{E}\{.\}$ is the expectation operator.

II. System Model

A. THz Communications Using Codebooks

In this paper, we consider the beam training between a base station (BS) and a user equipment (UE) that communicate

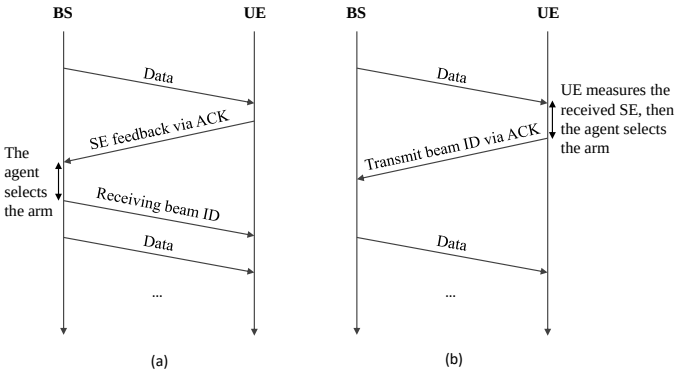

Fig. 1. The beam training protocols when the agent is implemented at: (a) BS; and (b) UE.

using antenna arrays. Let the number of antennas at BS and UE are N_{B} and N_{U}, respectively and $\mathbf{H}$ denote the complex channel matrix between BS and UE, $\mathbf{H} \in \mathbb{C}^{N_{\mathrm{U}} \times N_{\mathrm{B}}}$. In addition, we denote the codebooks for BS as $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_{M_{\mathrm{B}}}\}$ and for UE as $\mathcal{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{M_{\mathrm{U}}}\}$, where M_{B} and M_{U} are the codebook size equipped at BS and UE, respectively. Note that $\mathbf{f}_i \in \mathbb{C}^{N_{\mathrm{B}} \times 1}$ and $\mathbf{q}_i \in \mathbb{C}^{N_{\mathrm{U}} \times 1}$. Considering a scenario where BS utilizes the beamforming vector $\mathbf{f}_i$ while UE utilizes the beamforming vector $\mathbf{q}_j$, the instantaneous received signal at UE during the timeslot t, can be written as

$$y_{ij}(t) = \mathbf{q}_j^{\mathrm{H}} \mathbf{H}(t) \mathbf{f}_i s(t) + \mathbf{q}_j^{\mathrm{H}} \mathbf{n}(t), \tag{1}$$

where $s(t)$ and $\mathbf{n}(t)$ are the transmitted signal and noise vector at UE in the timeslot t. Hence, the instantaneous signal-to-noise (SNR) ratio at UE can be expressed as

$$\mathrm{SNR}(t) = \frac{\left|\mathbf{q}_j^{\mathrm{H}} \mathbf{H}(t) \mathbf{f}_i\right|^2}{\left|\mathbf{q}_j^{\mathrm{H}} \mathbf{n}(t)\right|^2}. \tag{2}$$

It can be seen that the combination of each $\mathbf{f}_i, \mathbf{q}_j$ pair results in varying SNR values. Thus, the objective is to identify the optimal beamforming pair that can achieve maximum long-term spectral efficiency (SE). The instantaneous SE of the link between BS and UE in a time slot t can be represented as:

$$\mathrm{SE}(t) = \log_2\left[1 + \mathrm{SNR}(t)\right]. \tag{3}$$

B. General MAB Platform For Beam Alignment And Tracking

In the 5G standard, when a new UE is added to the network for the first time, it waits for BS to initiate the Initial Access (IA) process [11]. During the IA procedure, BS transmits synchronization signals (SS) and UE measures the beam quality and report them back via ACK/NACK messages. The IA is periodically performed by BS to detect new UEs and update existing UE's best beams. To maintain communication, BS transmits Channel State Information Reference Signals (CSI-RS) at regular intervals. These signals are used for Reference Signal Received Power (RSRP) measurements for beam management during mobility. However, transmitting the CSI-RS towards all spatial directions may cause a long delay. To

TABLE I
ACTIONS AND MEANING

Actions	Meaning
a_0	BS uses beam 0 and UE uses beam 0
a_1	BS uses beam 0 and UE uses beam 1
a_2	BS uses beam 0 and UE uses beam 2
$\cdots$	$\cdots$
$a_{n_{\text{actions}}-1}$	BS uses beam M_{B} and UE uses beam M_{U}

support ultra-reliable low-latency communication (URLLC), it is necessary to significantly decrease the control overhead of beam training because THz communications will use a larger antenna array compared to the mmWave communications.

The problem of beam training can be formulated as a typical MAB learning problem, since the selection of any transmit and receive beam does not impact the environment's state. In this context, an agent must balance the acquisition of new knowledge, or *exploration*, with the optimization of decisions based on previously acquired knowledge, or *exploitation*. The objective is to optimize the trade-off between exploration and exploitation so that its total reward over a given period of time is maximized. The MAB has enjoyed widespread applications in various practical domains, including healthcare, advertising, and others.

Considering a set of available actions[1] $\mathcal{A}$, where the agent and the environment interact sequentially over T rounds. At each round $t = 1, 2, \cdots, T$, the agent selects an action $A_t \in \mathcal{A}$ to perform, and the environment returns a corresponding reward X_t to the agent. The interaction between the agent and environment provide a measure on the sequence of outcomes $A_1, X_1, A_2, X_2, ..., A_n, X_n$. The objective of the agent is to maximize the total reward $S_T = \sum_{t=1}^{T} X_t$, over the entire duration of the interaction. This goal is achieved by strategically selecting actions that lead to the highest possible cumulative reward.

Regarding beam training, the set of actions can be denoted as $\mathcal{A} = \{a_i\}$ with $0 \leq i < M_{\text{B}} M_{\text{U}}$. A mapping between each action and a pair of beams is shown in Table. I. Hence, there is a total of $n_{\text{actions}} \triangleq M_{\text{B}} M_{\text{U}}$ actions. At the time slot t, the reward can be defined as $X_t = \text{SE}(t)$. The agent is responsible for selecting the action in each time slot as UE moves, enabling online learning and adaptation. Unlike traditional supervised learning methods that require a training phase specific to each environment, MAB learning eliminates the need for offline training, resulting in reduced time and effort.

It is noteworthy that the agent can be implemented at either BS or UE, albeit with some required modifications to the relevant protocols. Fig. 1 demonstrates the beam training protocols for both scenarios, when the agent is at BS and when it is at UE. In the case when the agent is implemented at BS, following by each primary data transmission, UE needs to report the instantaneous SE value to BS via the ACK message. Subsequently, based on that value, the agent selects the action to play in the current time slot, following by BS sending the receive beam ID to UE. After this, the primary data communication will be initiated. In comparison, when the agent is located at UE, after each primary data transmission, the agent can make a decision of which action to play owing to the availability of the instantaneous SE value at UE. Subsequently, UE simply transmits these information to BS via ACK message, which initiates the next primary data transmission.

With this general platform for utilizing MAB learning for beam training, the agent can employ various MAB learning algorithms, which will be further elaborated in the subsequent sections.

[1]In the context of MAB learning, the terms *actions* and *arms* can be used interchangeably. This paper utilizes both terms to refer to the same concept.

Algorithm 1: UCB Algorithm [12]

Input: The uncertainty probability δ_1.
Output: A_t.

Initialize $\hat{\mu}_{t,a} = 0$ and $n_{t,a} = 0$.
for $t = 1, 2, \cdots$ **do**
  foreach $a \in \mathcal{A}$ **do**
    /*Calculate UCB for each action*/ $\text{UCB}_{t,a} = \begin{cases} +\infty, & \text{if } n_{t,a} = 0 \\ \hat{\mu}_{t,a} + \sqrt{\frac{2\log(1/\delta_1)}{n_{t,a}}}, & \text{otherwise} \end{cases}$
  end
  /* Select the action whose the highest UCB value */ $A_t = \arg\max_{a \in \mathcal{A}} \text{UCB}_{t,a}$
  /* Update the values after receiving the reward X_t */ $\mu_{A_t} \leftarrow \frac{\mu_{A_t} n_{t,A_t} + X_t}{n_{t,A_t}+1}$; $n_{t,A_t} \leftarrow n_{t,A_t} + 1$
end

III. Beam Training And Beam Tracking Via Bandit Learnings

A. Upper Confidence Bound Algorithm

The Upper Confidence Bound (UCB) algorithm is a widely employed strategy in the field of MAB problems [12]. It operates based on the principle of *optimism in the face of uncertainty*. This means that the agent should adopt an optimistic approach and act as if the environment is as favorable as plausibly possible. In the context of MAB, the UCB algorithm utilizes this principle to balance exploration and exploitation by selecting the action with the highest upper confidence bound, which incorporates both the empirical reward and an uncertainty term, defined mathematically as

$$\text{UCB}_{t,a} = \begin{cases} +\infty, & \text{if } n_{t,a} = 0 \\ \hat{\mu}_{t,a} + \sqrt{\frac{2\log(1/\delta_1)}{n_{t,a}}}, & \text{otherwise.} \end{cases} \tag{4}$$

Herein, $\text{UCB}_{t,a}$ stands for the UCB value for the action $a \in \mathcal{A}$, which is computed based on the expected reward of the action, denoted as $\hat{\mu}_{t,a}$, and the number of times the action has been

Algorithm 2: Linear Contextual UCB Algorithm [13]

Input: Probability of uncertainty δ_2. Set $\alpha = 1 + \sqrt{\ln(2/\delta_2)/2}$

Output: A_t.

for $t = 1, 2, \cdots, n_{\text{actions}}$ **do**
  Observe the contextual vector $\mathbf{x}_t \in \mathbb{R}^d$; **foreach** $a \in \mathcal{A}$ **do**
    if a *is new* **then**
      $\mathbf{A}_a \leftarrow \mathbf{I}_d$;
      $\mathbf{b}_a \leftarrow \mathbf{0}_{d\times 1}$;
    end
    /* Estimate the coefficient vector for each action a */ $\hat{\boldsymbol{\theta}}_a \leftarrow \mathbf{A}_a^{-1}\mathbf{b}_a$;
    /* Calculate the UCB for each action */ $\text{UCB}_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^{\text{T}}\mathbf{x}_{t,a} + \alpha\sqrt{\mathbf{x}_{t,a}\mathbf{A}_a^{-1}\mathbf{x}_{t,a}}$;
  end
  /* Select the action whose the highest UCB value */ $A_t = \arg\max_{a\in\mathcal{A}} \text{UCB}_{t,a}$
  /* Update the values after receiving the reward X_t */ $\mathbf{A}_{A_t} \leftarrow \mathbf{A}_{A_t} + \mathbf{x}_{t,A_t}\mathbf{x}_{t,A_t}^{\text{T}}$;
  $\mathbf{b}_{A_t} \leftarrow \mathbf{b}_{A_t} + X_t\mathbf{x}_{t,A_t}$;
end

selected, represented by $n_{t,a}$, all in a specific time slot t. This equation was proposed based on the following inequality [12]

$$\mathbb{P}\left(\mu_{t,a} > \hat{\mu}_{t,a} + \sqrt{\frac{2\log(1/\delta_1)}{n_{t,a}}}\right) \leq \delta_1, \tag{5}$$

for all probability of uncertainty $\delta_1 \in [0, 1]$. Generalling speaking, UCB is the greatest value of the reward that the agent expects. As we can see, when the value of $n_{t,a}$ is low, the square root term in the UCB formula becomes large, which results in a higher probability of exploration by the agent. In addition, if there is an action a which has not been played yet, its UCB will be set as $+\infty$, forcing the agent to play this arm for exploration.

The transmission protocol for UCB algorithm is demonstrated in Fig. 1.

B. Contextual Linear UCB Algorithm

Contextual bandits are a class of machine learning algorithms that take into account contextual information to select the best action depending on the contextual information [12]. The stochastic contextual bandits still operates based on the principle of optimism in the face of uncertainty, however, the agent exploits by modelling the relationship between the expected reward and the contextual vector, which is called reward model. In this paper, we adopt the linear function for the reward model, where the expected reward of each action

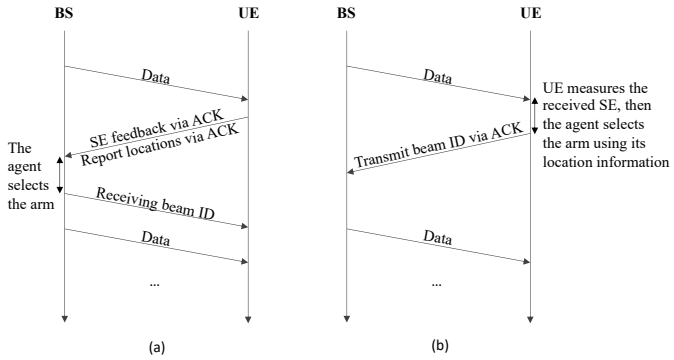

Fig. 2. The beam training protocols when the Loc-LinUCB algorithm is implemented at: (a) BS; and (b) UE.

(denoted as $\mathbb{E}[\text{X}_{t,a}]$) is modelled as a linear function of the contextual vector $\mathbf{x}_{t,a}$[2], which can be demonstrated as

$$\mathbb{E}[\text{X}_{t,a}|\mathbf{x}_{t,a}] = \mathbf{x}_{t,a}^{\text{T}}\,\boldsymbol{\theta}_a^*, \tag{6}$$

where $\boldsymbol{\theta}_a^*$ is an unknown coefficient vector $\boldsymbol{\theta}_a^*$ associated with the action a. These vectors are estimated when the agent receives rewards from the environment. In particular, there will be data associated with each action including the contextual vector and the corresponding reward. Based on this data, the coefficient vectors can be estimated online using Ridge regression [13]. The linear contextual bandit algorithm is described in Algorithm 2 [13]. Based on this, we will propose 2 algorithms, so-called Loc-LinUCB and Probing-LinUCB, which utilize different contextual vectors for beam training.

1) Linear Contextual Bandits With Location Data: Recent research has indicated a correlation between the positions of UEs and the Received Signal Strength Indicator (RSSI) [14]. These studies suggest that the geographical locations of UEs can serve as a valuable contextual vector, provided that the hardware of the UEs is capable of collecting this information. To leverage this fact, we introduce Loc-LinUCB, an algorithm that incorporates the contextual vector as a form of side information to help decision making. Specifically, the contextual vector is represented as $\mathbf{x}_{t,\text{loc}} = [x_t, y_t]^{\text{T}} \in \mathbb{R}^2$, where x_t and y_t refer to the x-coordinate and y-coordinate of UE, respectively.

The transmission protocol for UCB algorithm is demonstrated in Fig. 3.

2) Linear Contextual Bandits With Probing Information: Some studies have shown that consecutive beams at both BS and UE exhibit a spatial correlation [15]. Therefore, we propose an approach in which BS only utilizes a selected subset of beamforming vectors, known as *probing beams*, from its codebook for transmitting the RSs, instead of using all the available vectors. Similarly, UE can also utilize some selected beamforming vectors from its codebook. This technique has

[2]Note that in the context of beam training, all actions have a similar contextual vector in some cases. As such, we denote the contextual vector in the time slot t as $\mathbf{x}_t$.

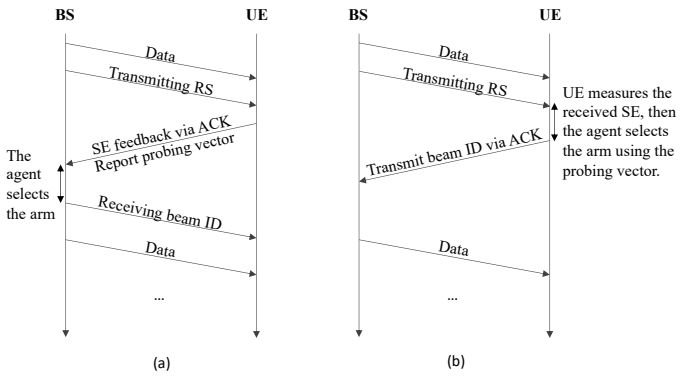

Fig. 3. The beam training protocols when the Probing-LinUCB algorithm is implemented at: (a) BS; and (b) UE.

the potential to significantly reduce the latency associated with RSs transmission. In this approach, the contextual vector refers to the sequence of SE values obtained after the RS transmission. We leverage this contextual vector to introduce a Probing-LinUCB algorithm that selects the optimal beams for each time slot.

The transmission protocol for UCB algorithm is demonstrated in Fig. 3. As we can see, compared to Loc-LinUCB, Probing-LinUCB requires some more sub-timeslots for sending the RSs, which will lead to a prolong latency.

IV. Performance Evaluation

A. Simulation Setup

In this paper, we benchmark the performance of the three algorithms, UCB, Loc-LinUCB and Probing-LinUCB using data generated by the DeepMIMO framework [10]. The framework employs precise ray-tracing data from Remcom Wireless InSite [16] to create a wireless channel that takes into account the influence of the environment geometry, as well as the transmitter and receiver positions. Specifically, we utilize the ray-tracing scenario O1 at 140 GHz as a benchmark, which simulates an outdoor setting with two streets and one intersection, as depicted in Figure 4. By conducting this benchmark study, we aim to shed light on the strengths and weaknesses of the three algorithms under consideration, and provide insights into their suitability for practical applications in THz communications.

In practice, DFT codebooks are widely used as they can match approximately the optimal beamforming. In addition, they also can achieve higher antenna gains at the beam directions than the codebooks used in IEEE 802.15.3c. Assuming N is the number of antennas and M is the number of beam patterns, a codebook $\mathbf{W}$ is defined as [17]

$$\mathbf{W}(n,m) = \frac{1}{\sqrt{N}} e^{-m2\pi \frac{nm}{N}}, \tag{7}$$

with $0 \leq n < N-1, 0 \leq m < M-1$. In this paper, both BS and UE employ DFT codebook with a resolution of 3 bits and a codebook size of $M_{\mathrm{B}} = 8$ and $M_{\mathrm{A}} = 4$, respectively.

TABLE II
SYSTEM PARAMETERS

Parameters	Value
Operating frequency (f_c)	140.0 GHz
Bandwidth (B)	0.5 GHz
BS antenna array	$1 \times 4 \times 4$ UPA
UE antenna array	$1 \times 2 \times 2$ UPA
Codebook size at BS (M_{B})	8
Codebook size at UE (M_{U})	4
Antenna spacing (d)	Half of a wave-length
Transmit power of BS (P)	10.0 Watts
Number of paths	5
Noise figure	7.2 dB

Regarding BS, we only use the BS 5 from the dataset for the simulations. To be specific, BS has a height of 6 meters and is equipped with a 4×4 uniform plannar array (UPA) antenna. The transmit power of BS is 10 Watts.

Regarding UE, it has a height of 2 meter above the ground and is equipped with a 2×2 uniform plannar array (UPA) antenna. Unlike existing research, we consider a continuous movement of UE to evaluate the beam training performance. The trajectory of UE is illustrated in Fig. 4, starting from point A and moving sequentially to points B, C, D, E, F, G, H, and finally returning to A. As UE moves, it is expected to experience line-of-sight (LOS) channels to BS at different locations, such as during the movement from A to B to C, while also experiencing non-LOS channels at other locations. Therefore, the bandit algorithms must adapt to the changes of the environment and make decisions accordingly.

Other system parameters are listed in Table II. In this section, we will provide the benchmark of the SE achieved under the following algorithms:

- *UCB algorithm* with $\delta_1 = 0.05$.
- *Loc-LinUCB algorithm* with $\delta_2 = 0.05$.
- *Probing-LinUCB algorithm* with $\delta_2 = 0.05$. The contextual vector is the sequence of instantaneous achieved SE when the actions a_0, a_8, a_{16}, a_{24} were utilized for transmitting reference signals. This means that only 4 sub-timeslots are spent on the RS transmission, instead of 32 as used by an exhaustive search.
- *Exhaustive search*, or *optimal algorithm*, in which the optimal pair of beamforming vectors are determined by using exhaustive search over the entire beam space.

B. Performance Evaluation

Fig. 5 illustrates the average SE achieved by different algorithms, namely UCB, Loc-LinUCB, Probing-LinUCB, and exhaustive search, when UE follows its designated path. It is worth noting that the SE values are averaged over every 100 timeslots. The results indicate that the performance of these algorithms fluctuates significantly depending on the location of UE. For instance, as UE moves towards BS, the SE increases steadily from timeslot 0 to 1500. However, abrupt changes in performance occur when UE changes direction. Interestingly, even when the exhaustive search algorithm is used, there are still significant fluctuations in performance observed during timeslots 4000 to 5000 (around 10 bits/s/Hz) and 5200 to 6200

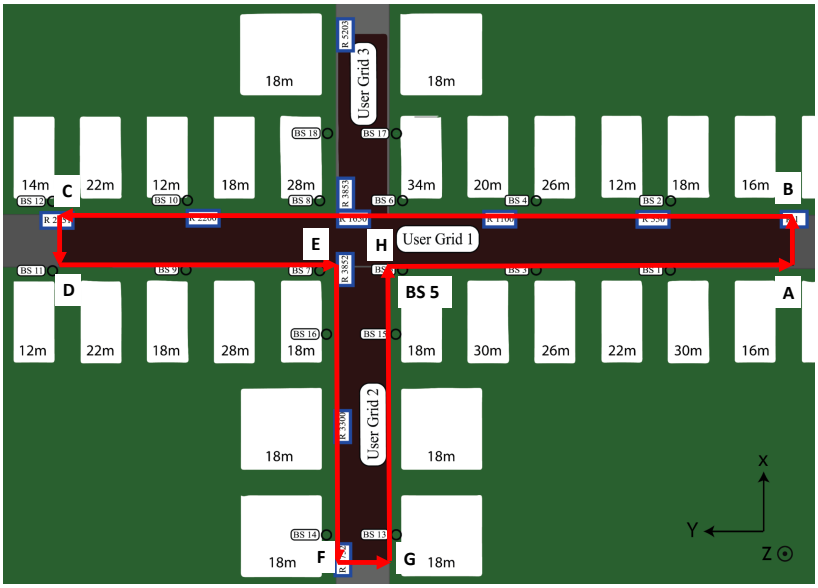

Fig. 4. The figure shows the top view of the O1 ray-tracing scenario and highlights the path taken by UE.

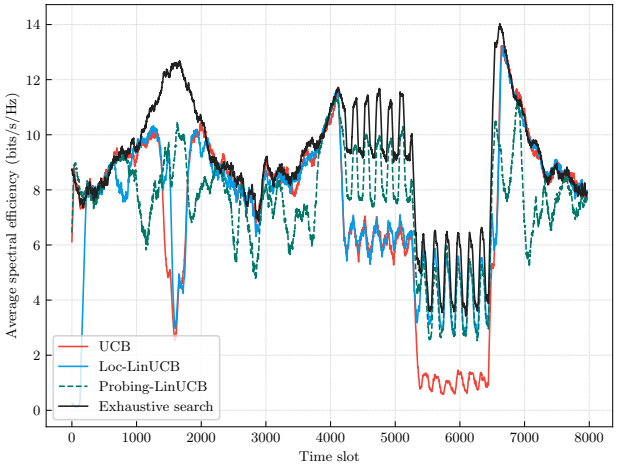

Fig. 5. The figure shows the average SE achieved by UCB, Loc-LinUCB, Probing-LinUCB and using exhaustive search when UE moves.

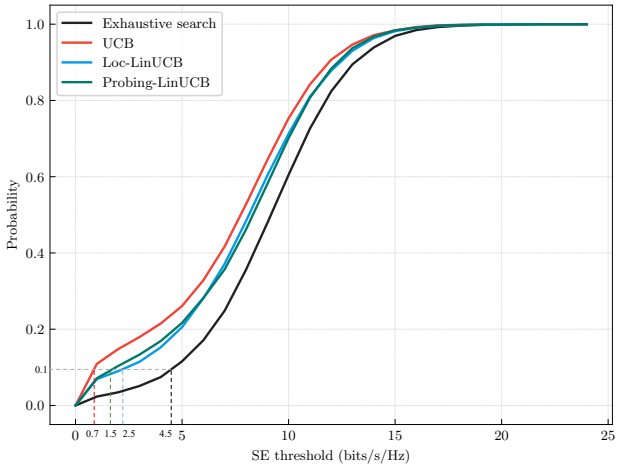

Fig. 6. The figure shows the CDF of the spectral efficiency achieved by UCB, Loc-LinUCB, Probing-LinUCB and using exhaustive search.

(around 4.5 bits/s/Hz). In the case when UE changes direction, the statistical distribution of the reward function undergoes an abrupt shift, leading to an suboptimal performance. As a result, all the algorithms must spend multiple timeslots for exploring the best action, as observed in the UCB algorithm's behavior between timeslots 1500 and 2000. During the exploration, a suboptimal performance can be observed. Despite achieving nearly optimal performance between timeslots 2000 and 4000 (e.g., about 8 bits/s/Hz), and again from 6500 to 8000 (e.g., approximately 10 bits/s/Hz), the UCB algorithm's performance is notably low between timeslots 4000 and 5000 (approximately 6 bits/s/Hz) and 5200 to 6000 (approximately 1 bits/s/Hz). This is because the fast environmental changes does not allow the agent to perform a sufficient exploration, hindering its ability to make appropriate decisions. In other words, the environment has changed into a new state before the agent can find out the best action for that state. In comparison, Probing-LinUCB outperforms UCB and Loc-LinUCB in the time slots around 1500 and between 4200 to 5200, while Loc-LinUCB performs better in the time slots between 5400 and 6400. This indicates that the use of a linear function to represent the relationship between the expected rewards and the contextual vector can lead to good performance in some cases, but may fail to generalize in other cases. This is evident in the performance of Probing-LinUCB in the time slots between 2000 to 4000, where it shows unpredictable fluctuation in the suboptimal performance.

Fig. 6 displays the cumulative density function (CDF) of the SE achieved by UCB, Loc-LinUCB, Probing-LinUCB, and

TABLE III
COMPARISON OF ALGORITHMS

	UCB	**Loc-LinUCB**	**Probing-LinUCB**
Complexity	Lowest	Higher	Higher
Extra hardware	No	Yes	No
Spend sub-timeslots for RSs	No	No	Yes
SE performance	Lowest	Runner-up	Best

exhaustive search. The results indicate that UCB achieves the lowest performance among all algorithms. On the other hand, both Loc-LinUCB and Probing-LinUCB show improved performance by incorporating side information into the decision-making process. Generally, these two algorithms demonstrate quite similar performance. However, it is noteworthy that for a probability of 0.1, the SE threshold for UCB, Probing-LinUCB, and Loc-LinUCB is approximately 0.7, 1.5, and 2.5 bits/s/Hz, respectively. However, its performance is still approximately 40% lower than the exhaustive search method (4.5 bits/s/Hz), primarily because the algorithm needs to perform exploration again when the environment changes.

Discussions: In practical applications, it is important to consider the strengths and weaknesses of the three algorithms. For instance, although the UCB algorithm exhibits the lowest performance among the three, it has a simple and straightforward implementation that is well-suited for hardware with low capability. In contrast, the Loc-LinUCB algorithm delivers an improved performance compared to UCB, but requires additional hardware to acquire location information for UE. Additionally, neither the UCB nor the Loc-LinUCB algorithms require extra sub-timeslots for transmitting RSs, which is a requirement for the Probing-LinUCB algorithm. It is worth highlighting that the complexity of both Loc-LinUCB and Probing-LinUCB is heavily dependent on the length of the contextual vector, despite being built upon LinUCB. This emphasises the need to carefully consider the dimensionality of the input data when implementing these algorithms in practice. A summary of these characteristics can be found in Table III.

V. CONCLUSIONS AND FUTURE RESEARCH

This paper presents three algorithms, namely UCB, Loc-LinUCB, and Probing-LinUCB, which adopt MAB learning in beam training at THz frequency. UCB is designed based on Upper Confidence Bound algorithm whereas Loc-LinUCB and Probing-LinUCB rely on contextual bandit learning. The results show that Loc-LinUCB and Probing-LinUCB outperform UCB thanks to utilizing contextual information for decision-marking process. There are several interesting research directions that can be pursued in the future. Firstly, given the wireless channel's tendency to undergo sudden and unpredictable changes, the data captured by the UE in the distant past may no longer be relevant in predicting current patterns. As a result, it is critical to design a contextual algorithm that can effectively adapt to such dynamic changes in the environment. Secondly, a strategy for selecting the optimal patterns to transmit probing beams needs to be investigated. Lastly, since the Loc-LinUCB and Probing-LinUCB algorithms each possess unique strengths, a natural question arises: how can these two algorithms be combined to further enhance system performance? Finding an effective way to incorporate the strengths of both algorithms holds significant promise for improving overall spectral efficiency.

REFERENCES

[1] I. F. Akyildiz, C. Han, Z. Hu, S. Nie, and J. M. Jornet, "Terahertz band communication: An old problem revisited and research directions for the next decade," *IEEE Trans. Commun.*, vol. 70, no. 6, pp. 4250–4285, 2022.

[2] B. Ning, Z. Tian, W. Mei, Z. Chen, C. Han, S. Li, J. Yuan, and R. Zhang, "Beamforming technologies for ultra-massive MIMO in terahertz communications," *IEEE Open J. Commun. Soc.*, 2023.

[3] F. Lemic, S. Abadal, W. Tavernier, P. Stroobant, D. Colle, E. Alarcón, J. Marquez-Barja, and J. Famaey, "Survey on Terahertz nanocommunication and networking: A top-down perspective," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 6, pp. 1506–1543, 2021.

[4] S. Dinh-Van, T. M. Hoang, R. Trestian, and H. X. Nguyen, "Unsupervised deep-learning-based reconfigurable intelligent surface-aided broadcasting communications in industrial iots," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 19 515–19 528, 2022.

[5] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37 328–37 348, 2018.

[6] A. Klautau, N. González-Prelcic, and R. W. Heath, "Lidar data for deep learning-based mmWave beam-selection," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 909–912, 2019.

[7] M. Hashemi, A. Sabharwal, C. E. Koksal, and N. B. Shroff, "Efficient beam alignment in millimeter wave systems using contextual bandits," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 2393–2401.

[8] I. Chafaa, E. V. Belmega, and M. Debbah, "One-bit feedback exponential learning for beam alignment in mobile mmWave," *IEEE Access*, vol. 8, pp. 194 575–194 589, 2020.

[9] A. Asadi, S. Müller, G. H. Sim, A. Klein, and M. Hollick, "Fml: Fast machine learning for 5g mmwave vehicular communications," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1961–1969.

[10] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," in *Proc. of Information Theory and Applications Workshop (ITA)*, San Diego, CA, Feb 2019, pp. 1–8.

[11] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, "A tutorial on beam management for 3GPP NR at mmWave frequencies," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 1, pp. 173–196, 2018.

[12] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[13] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 661–670.

[14] R. Deng, S. Chen, S. Zhou, Z. Niu, and W. Zhang, "Channel fingerprint based beam tracking for millimeter wave communications," *IEEE Commun. Lett.*, vol. 24, no. 3, pp. 639–643, 2019.

[15] Y. Heng, J. Mo, and J. G. Andrews, "Learning probing beams for fast mmWave beam alignment," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 1–6.

[16] Remcom, "Wireless InSite," 2023, http://www.remcom.com/wireless-insite.

[17] S. Dinh-Van, T. M. Hoang, B. B. Cebecioglu, D. S. Fowler, Y. K. Mo, and M. D. Higgins, "A defensive strategy against beam training attack in 5G mmWave networks for manufacturing," *IEEE Trans. Inf. Forensics Secur.*, pp. 1–1, 2023.